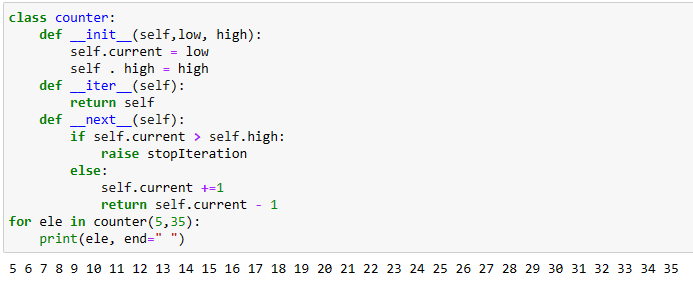
**Q1. Which two operator overloading methods can you use in your classes to support iteration?**

**Ans: \_\_iter\_\_ and \_\_next\_\_ are the operator overloading methods in python that support iteration and are collectively called iterator protocol.**

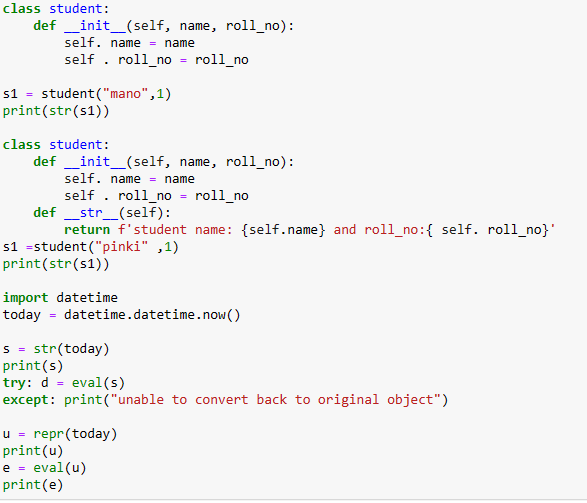
* **\_\_iter\_\_ returns the iterator object and is called at the start of loop inour respective class.**
* **\_\_next\_\_ is called at each loop increment , it returns the incremented value. Also stop iteration is no value to return.**

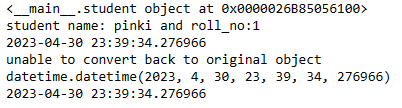
****

**Q2. In what contexts do the two operator overloading methods manage printing?**

Ans: \_\_str\_\_ and \_\_repr\_\_ are two operator overloading methods that manage printing .

* The difference between both of these operator is : The goal of \_\_repr\_\_ is to be unambiguous and \_\_str\_\_ is to be readable.
* Whenever we are printing any object reference internally , \_\_str\_\_ method will be called by default.
* The main purpose of \_\_str\_\_is for readability . it prints the informal string representation of an object . it may not be possible to convert result string to original object.
* \_\_repr\_\_ is used to print official string representation of an object , so it includes all information and development.





**Q3. In a class, how do you intercept slice operations?**

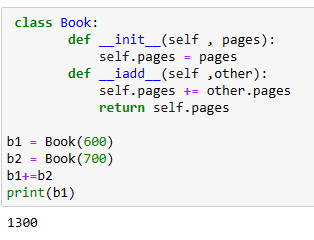
**Ans**: in a class use of **slice( )** , **\_\_getitem\_\_** method is used for intercept slice operation . This slice method is provided with start integer, stop integer number and step integer number.

Example:

**\_\_getitem\_\_(slice(start, stop, step))**

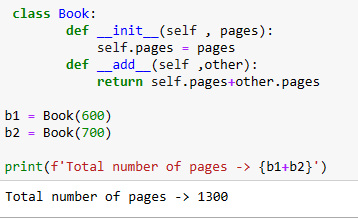
**Q4. In a class, how do you capture in-place addition?**

Ans: **a+b** is normal addition .whereas **a+=b** is inplace addition operation . in this in-place addition **a** itself will store the value of addition . in a class **\_\_iadd\_\_** method is used for this in-place operation



**Q5. When is it appropriate to use operator overloading?**

**Ans**: operator overloading is used when we want to use an operator other than its normal operation to have different meaning according to the context required in user defined function.

****